Significant Event System
Design Document

Geoff Savage

1 Overview

The significant event system (SES) is a DAQ tool that notifies users of situations where the data could be negatively affected. When an error is severe enough data acquisition is interrupted. A record of activity in the SES is maintained in a log file. Users can access the archive log file offline to review the detector's condition during the run. The system consists of a set of software components communicating in a predefined protocol through messages.

2 Software Components

The significant event system consists of the following software components.

- SE Server
- SE Display
- SE Logger
- SE Log scanner
- SE Watcher
- SE Database Server
- DAQ software applications
- SE Message Board
- SE Generator

2.1 SE Server

The significant event server is the heart of the significant event system. The server processes significant event messages. The only SES messages not processed by the server are hardware database messages between the SE Display and the SE Database Server. All other significant event messages are sent to the server and then distributed as requested or the server can initiate a SE message.

Applications in the D∅ data acquisition (DAQ) system register with the server to send, receive, or send and receive significant event messages. The other parts of the SES, display, watcher, and logger, are event receivers. Error messages that signal a condition requiring operator intervention are added to a list in the server and then sent to the receivers. Clearing an error condition sends a message stating that the system has returned to its normal operating state. Upon receipt of the alarm state cleared message the server updates its internal list and then sends the alarm state cleared message to the requestors. Clearing an alarm does not remove the alarm from the servers' internal list. The memory of the alarm condition is used to notify users that their actions have resolved the problem. Later, when the event is removed from the servers' internal list the server initiates a message notifying the display to remove the problem-resolved message. Notifying users that their actions resolved a problem gives them a warm fuzzy feeling.

Run stop messages are sent from the server to the SE Watcher for conversion to run control message format. Messages that provide information or warnings are received by the server and sent to receivers requesting these message types. These messages require no operator intervention.

All significant events are prioritized. Messages are sent to one queue in the server. Upon arrival the message is removed from the queue, time stamped, and placed in a queue corresponding to its priority. High priority messages are processed before low priority messages.

The SE Server is also the heartbeat monitor. Heartbeats insure that all the critical software tasks are running. Applications in the DAQ system that require monitoring tell the server that they will be sending messages periodically, a 'tic'. There is a minimum period. A period that is too short will swamp the server with heartbeat messages. If a 'tic' is not detected within the specified time the server signals an alarm that appears on the heartbeat screen of the SE Display. When a 'tic' from the application in alarm reappears, the server clears the alarm condition.

The SE Server maintains statistics of the significant events and connections it processes. Statistics are displayed using the expert interface to the server. The expert interface accesses the internals of the server directly. It is designed for use by experts to diagnose and resolve problems with the server.

2.2 SE Display

Users monitor the state of the experiment using a significant event display. Information displayed is user selectable.

2.3 SE Logger

The SE Logger writes significant events to an archive log file. It is a receiver client of the SE Server. Usually no filters are set for the logger. All the significant event messages received from sender clients by the SE Server are forwarded to the logger for archiving. Heartbeat tics are not logged but heartbeat configuration messages are logged. SE messages generated by the SE Server are also logged.

Without the logger executing the SES provides no record of significant events. For this reason the logger is a heart beater. If the server loses heartbeat tics from the logger the run is stopped.

The logger manages the archived log files. It monitors disk space to insure that there is enough room to write the files and segments the files into manageable portions. Log file size can be limited by extent or by DAQ runs. The logger also creates backups of log files at user selectable intervals.

The logger uses the ZOOM ErrorLogger interface for output to archive log files. Users use the SE Log Scanner to view the contents of a log file.

2.4 SE Log Scanner

Over the course of the experiment many significant events will be recorded in the log file. The significant event log scanner is a utility that aids in examining the log files.

The SE Log Scanner is a utility for perusing archive log files. It uses the ZOOM ErrorLogger interface access the significant event messages which is the interface the SE Logger uses to write to the log files. The log scanner uses a GUI to interface with the user.

Since log files can contain thousands of significant event messages the log scanner is able to filter the messages for display. The filtering capabilities match those of the server for forwarding SE messages to receiver clients. The filtered messages are displayed in a color-coded format that allows users to easily identify the message type.

The log scanner also generates reports. The report interface is through the ZOOM ErrorLogger. An entire file can be printed or the results of an applied filter can be printed.

2.5 SE Watcher

The SE Watcher communicates information to the run control software. The watcher is a receiver client. It requests all significant event messages from SE Server that signal the run needs to be stopped. The SE stop message is translated to a run control stop message and is sent to run control.

The SE Watcher is a heart beater. Without this process signaling through the significant event system can't stop the run.

2.6 Database Server

The Database Server interfaces between the SE Display and the hardware database. When an alarm is signaled the user has the option to view the values applicable to the alarm condition. The values include the alarm limits. The current limits are downloaded from the front end and the design limits are obtained from the hardware database. The Database Server converts data in the hardware database format to the SE Display format.

2.7 DAQ Software Applications

Significant event messages can be generated by:

1) VME Front ends
2) Host applications
    a) Physics event logging
    b) Physics event monitoring
    c) Run control
    d) Detector monitoring
3) Trigger processors

Each software application must embed SE generation code to be executed when the application identifies a significant event.

2.8 SE Message Board

The SE Message Board is an application designed for use in the SES development environment not the DAQ run environment. It is a receiver client of the server that displays the messages immediately in its' text window. This allows application developers to see the effects of sending a significant event immediately. The alternative is to use the SE logger and log scanner. Since the message board is just another receiver client it can be used with the other applications in the SES during a run if needed.

2.9 SE Generator

Generate generic SE event messages at user selectable intervals. Used for SES timing tests and statistics tests.

3 Communication

Communication within the SES describes how information flows between the participating applications. Applications in the SES communicate via messages sent over connections.

3.1 Connections

Before messages can be sent between applications the two applications must be connected. The connections are handled by D∅ME (the D∅ client/server messaging package). One application is identified as the server and the other the client. The SES uses one server (SE server); all other applications are clients. It is the client's responsibility to connect and maintain the connection to the server. When the client is finished it should disconnect from the server (Hopefully nicely). If the server crashes it is the client's responsibility to reconnect when the server is running again.

3.2 Messages

A message is a piece of information transferred between two software components of the SES. The messages apply to the following functions:

- Significant event
- Heartbeat
- Database
- Server requests

The client/server package, D∅ME, is used to transfer messages and maintain connections.

3.2.1 Significant Events

Applications generate significant events for three reasons:

- Alarm signaled - A system has moved from a good state to a bad state.
- Alarm canceled - A system has moved from a bad state to a good state.
- Information alarm - A system needs to report some information.

Applications must define their role in the SES. The roles are:

- Generator – send significant events from app to server
- Receiver – send significant events from server to app
- Generator and Receiver – send significant events in both directions

3.2.2 Heartbeats

Heartbeats monitor the activity of other processes. A process notifies the SES that it will deliver a message at regular intervals. If the message does not arrive the SES signals an alarm condition. Subsequently if the SES receives a heartbeat message then the alarm is turned off.

3.2.3 Database

The hardware database contains the baseline settings of channels in the DAQ system. DAQ settings can be modified after the values are loaded from the database. When an alarm occurs it is often helpful to compare the current and baseline settings. This is accomplished by sending a request message to the database server. Typically requests are sent from the SE Display application. Database values are returned to the requesting application in a message.

3.2.4 Server Requests

At times the SE server needs information from the clients.

- Regenerate alarm status. When SE server starts or if things get out of sync.
- Collection of statistics.
- Other data. (EPICS values, …)

3.3 Filters

Ability to control which significant events are sent to receivers. A message is passed on when all components of the filter specified are present in the message. Exceptions can also be specified. All messages except those matching the specified filter are passed on. Filters only apply to receivers and can be added or deleted at anytime. Filters are applied to all fields except text.

4 Data Flow

The basic data flow in the SES is shown in this simple diagram.

5 Message Specification

All SES messages are D∅ME string messages. This eliminates the need to worry about byte swapping on different CPU architectures and support already exists in Python (thanks to Scott Snyder). Each string is divided into words with the initial word in the string being referred to as word zero (0). This keeps a direct translation between the specification and the coding.

Word	Function
0	Version number
1	Type
2	Severity/Priority
3	Identifier
4	Text

5.1 Version Number

Identifies the version of the string message format. This allows for expansion while maintaining compatibility with earlier versions.

5.2 Type

Describes the functionality of the message. Each leg of the communication protocol performs a different function.

5.3 Severity/Priority

A measure of the messages' importance. EPICS prioritizes by severity and Run I prioritized by Priority (see D∅NOTE 1064). By using a string in this field both styles can be accommodated.

5.4 Identifier

Identifies the source of the message. The source is a colon delimited field that identifies the <subsystem>:<subsubsystem>:<subsusbsubsystem>. For an IOC (VME front end):

- subsystem = IOC IP name
- subsubsystem = slot number
- subsubsubsystem = channel number

All fields need not be specified but at least one field must be specified. In the case of EPICS this field contains the PV name for accessing values from the hardware database. See D∅NOTE 1098 for the Run I hardware database device names, subsystems, and hierarchical path designations.

5.5 Text

Any additional information the application wants to display, log, or transfer to other apps.

6 Commands

Outlines the commands used in the SES communication protocol.

6.1 Server Commands

Commands sent from clients to the server.

Category	Command	Description
Alarm	signal_alarm (add_alarm)	Good to bad.
	cancel_alarm	Bad to good.
Filter	add_filter	Specify filter pattern
	delete_filter	Remove filter pattern
	modify_filter	Modify filter pattern.
	filter	Dump filter message?
	send_filter	Send the filter pattern to the client
Event	send_event	Send all current and future event messages that match the filter.
	send_new_event	Send all future event messages that match the filter.

	keep_event	Discontinue sending messages.
	event	Dump event message?
	clear_event	?
Heartbeat	add_heartbeat	Server expects heartbeat tics.
	delete_heartbeat	Server no longer expects heartbeat tics.
	heartbeat_tic	Tell the server that you are alive.
	send_heartbeat	Send information on all current and future heart beat sources. Server also echo's heartbeat_tics.
	keep_heartbeat	Discontinue sending heartbeat tics and sources.
Monitor	one_monitor	Send internal information once.
(Statistics)	send_monitor	Send internal information periodically.
	keep_monitor	Stop sending internal information.
Connection	send_connection	Send info on current and future connections.
(Circuit)	keep_connection	Stop sending connection info.
Server	reset_system	Start over
Time	time	Dump time
	send_time	Send server time
	keep_time	Stop sending server time
Run Control	begin_end	Dump begin/end message
	send_begin_end	Send run start/stop messages
	keep_begin_end	Stop sending start/stop messages

6.2 Client Commands

Commands sent from the server to clients.

Category	Command	Description
Connection	disconnect	Connection with SE server has been terminated.
	disconnect_warning	Connection with SE server will be terminated shortly.
Alarm	See server commands	
Heartbeat	See server commands	
	lost_heartbeat	Did not receive heartbeat tic in time window
	found_heartbeat	Received heartbeat tic after being lost
Protocol	msg_received	Acknowledge receipt of message.
	msg_processed	Acknowledge processing of message.

7 Message Protocol

The SES message protocol defines the timing of message passing between two applications in the SES.

7.1 Connect

1. Establish a connection with the significant event server
2. Remember d∅me connection identifier
3. Register callback for disconnect

4. Register callback to handle messages from the SE server

7.2 Disconnect

1. NULL connection identifier in disconnect callback
2. Unregister callback for disconnect
3. Unregister callback to handle messages from the SE server
4. Attempt to reconnect

7.3 Significant Events

7.3.1 Generate

1. Inform server that app is a generator
2. Send client id information
3. Send messages as needed
4. Inform server that app is closing connection

7.3.2 Receive

1. Inform server that app is an event receiver
2. Send client id information
3. Process received messages as needed
4. Inform server that app is closing connection

7.3.3 Generate and receive

1. Inform server that app is a generator/receiver
2. Send client id information
3. Send messages as needed
4. Inform serve that app is closing connection

7.4 Heartbeats

Heartbeats monitor the activity of other processes. A process notifies the SES that it will deliver a message at regular intervals. If the message does not arrive the SES signals an alarm condition. Subsequently if the SES receives a heartbeat message then the alarm is turned off.

7.4.1 Generate

1. Inform server that app will send heartbeats
2. Request minimum interval from server
3. Start heartbeater with at least minimum interval
4. Inform server that app will no longer send heartbeats

8 Application Internals

Describe the internal actions prompted by messages arriving or internal events for each application.

8.1 SE Server

8.1.1 Statistics

The server maintains statistics on the messages received. The statistics aid in characterization of system response and debugging problems. This section outlines the information tabulated to achieve these goals and how to access this information.

8.1.2 Expert Interface

The expert interface in Run I supported these commands.

Command	Action
HELP	Print this list
ACTIVITIES	Show current activities
RECONNECT	Reconnect to the SE Server
LOCAL MONITOR	Show local monitors
CIRCUITS	Show circuits
ONE CIRCUIT	Single circuit
FILTER	Show filters on a circuit
MONITORS	Show monitors
ONE MONITOR	Single monitor
REPORT	
NEW EVENTS	Show new events
EVENTS	Show events
ACKNOWLEDGE	Acknowledge event
UNACKNOWLEDGE	Un-Acknowledge event
HEARTBEATS	Show heartbeats
ADD FILTER	Add a filter
DEL FILTER	Delete a filter
BEGIN	Send/Keep Begin/End events
TIME	Send/Keep Time events
APING	Asynchronous ping of SE Server
SPING	Synchronous ping of SE Server
VERBOSE	Make verbose dumps
RESET NODES	Tell CDAQ to reset alarms on front end node(s)
CLOSE LOG	Close SE Server log file
SET MON PER	Set SE Server monitor period
SET RTR PER	Set alarm server retry period
CHECK CIRC	Check connections on all circuits
DISCON CIRC	Forcibly disconnect a single circuit
ABORT CIRC	Forcibly abort a single circuit
ADD HEARTBEAT	Create a heartbeat
DEL HEARTBEAT	Delete a heartbeat
SHUTDOWN	Shutdown alarm server
ABORT	Abort alarm server
RESET SYS	Clear event queue for single system

CLEAR EVENT	Clear entire event queue
CONNECT G	Force connection to Gateway
DISCONNECT G	Force disconnection from Gateway
SET ACTIVE	Set SE server to be active
SET INACTIVE	Set SE server to be inactive
EXIT	Exit server control program
Negative prefix stops the current activity	

8.2 SE Display

8.3 SE Logger

8.4 SE Log scanner

8.5 SE Watcher

8.6 Heartbeat Monitor

8.7 Database Server

8.8 C++ Applications

8.9 VxWorks Applications

9 Glossary

Term	Definition
CDAQ	Run I data acquisition control
Circuit	DECNET connection between two applications
Gateway	Bridge between DECNET network and token ring network